



Sheet #1 (Pointers)

- Are each of the following declarations valid or invalid? If any are invalid, why?
 - `int ivar;`
`int *iptr = &ivar;`
 - `int ivar, *iptr = &ivar;`
 - `float fvar;`
`int *iptr = &fvar;`
 - `int nums[50], *iptr = nums;`
- What is the output of the following program assuming array is an array of ints:

```
if (&array[4] < &array[1])
    cout<< "True";
else
    cout<< "False";
```
- Assume `pint` is a pointer variable. Are each of the following statements valid or invalid? If any are invalid, why?
 - `pint++;`
 - `pint += x; // Assume x is an int`
 - `pint /= 2;`
 - `pint *= 4;`
- What is wrong in the following program:

```
int numbers[4] = {10, 20, 30, 40, 50};
cout<< "The third element in the array is ";
cout<< *numbers + 3 << endl;
```
- What is wrong in the following program:

```
int *pint;
pint = new int[100];    // Allocate memory
cin>>pint;
delete pint;           // Free memory
```
- True or false:**
 - Any mathematical operation, including multiplication and division, may be used on a **pointer**.
 - To calculate the **amount of memory** used by an array, multiply the number of elements by the number of bytes each element uses.
 - A pointer that contains address 0 is called a **null pointer**.
 - When you **add a value to a pointer**, you are actually adding that number to the value stored in the address stored in the pointer.



Sheet #1 (Pointers) Cont.

7. Trace the following codes and write the output for each:

- a. `#include <iostream.h>`
`void main ()`
`{`
`int a=2, b=3;`
`int *z;`
`z = &a ;`
`a = b ;`
`cout<<" value 1 = "<< a << "\n at address " << &a << endl;`
`cout<<" value 2 = "<< b << "\n at address " << &b << endl;`
`cout<<" value 3 = "<< *z << "\n at address " << z << endl;`
`}`
- b. `#include <iostream.h>`
`void duplicate (int a, int *b, int *c)`
`{`
`a=2;`
`*b=2;`
`*c=2;`
`}`
`void main ()`
`{`
`int x=1, y=3, z=7;`
`duplicate (x, &y, &z);`
`cout<<"x= "<<x<<"\t y= "<<y<<"\t z= " << z<<endl;`
`}`
- c. `#include <iostream.h>`
`void main ()`
`{`
`int a[] = {1, 2, 4, 6, 12, 3, 9};`
`int *z;`
`int i;`
`z = a ;`
`for(i=0; i<7; i++)`
`{`
`cout<< *z<<endl ;`
`z++;`
`}`
`}`



Sheet #1 (Pointers) Cont.

- d. `#include <iostream.h>`
`void main ()`
`{`
`float table[]={ 1.1,2.3,4.5,6.7,8.7,6.6,4.0,3.3,2.7,4.5};`
`float *pt, *qt;`
`pt=table;`
`*pt=0;`
`qt=pt+2;`
`*(pt+2)=3.14;`
`cout<<table[0]<<"\t" << table[2]<< "\t"<< *pt <<"\t"<< *qt<<endl;`
`}`
- e. `#include <iostream.h>`
`void main ()`
`{`
`int a[] = {1, 2, 4, 6, 12};`
`char c[] = "Hello";`
`int *z;`
`char *x;`
`int i;`
`z = a;`
`x = c;`
`for(i=0; i<5; i++)`
`{ cout<<*z<<"\t"<< *x<<endl ;`
`*z++;`
`x++;`
`}`
`}`
- f. `#include <iostream.h>`
`int SomeFunction(int *m)`
`{`
`*m=(*m)+1;`
`return(*m);`
`}`
`void main ()`
`{`
`int *a,b=1,c=2,d=3,*e;`
`a=&d;`
`e=&c;`
`*a=1;`
`cout<<*e<<"\t"<< c<<"\t"<< d<<endl;`
`b=someFunction(&c);`
`cout<<c<<"\t"<<*e<<end;`
`}`